Remarks/Arguments

The Office has rejected claims 14 and 17-19 under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention. In addition, the Office has rejected all claims (claims 1-29) under 35 U.S.C. 102(b) as being anticipated by Chow *et al.* (U.S. Patent 5,875,334). In light of the arguments below, Applicant asks the Office to reconsider these rejections and to allow all of the claims.

The 112 Second Paragraph Rejection

Amendments to claims 14 and 17 addressing the 35 U.S.C. 112, second paragraph, rejections are shown in the listing of claims above. As shown, claim 14 is amended by removing the redundant limitation that the "database query language statement is not one of the predetermined types of expressions," consistent with the Office's comment. In addition, a second, extraneous period was also removed from the end of claim 14.

As shown, claim 17 is amended to address the 35 U.S.C. 112, second paragraph, rejection by having the database system evaluate a database query language statement in the second type expression when evaluating the second type expression, rather than having the database system submit a database query language statement to the database system, thereby eliminating the confusion suggested by the Office to exist in the original. Claims 18 and 19, being rejected on the basis of the claim 17 rejection, are left as originally submitted.

The 102(b) Rejection in view of Chow *et al.*

With respect to claim 1, Chow does not teach generating "object code" to represent the stored procedure, as required in Applicant's amended claim 1. Rather, in Chow, threaded – not object – code is generated to represent both the control expression ("procedural") and query statement ("non-procedural") parts. As is well known to one skilled in the art, threaded code requires an interpreter to be executed. The object code claimed by Applicant does not require an interpreter. This distinction is significant, as interpreting threaded code is a slower, and thus less efficient, process than is executing object code. In particular, each statement in a piece of threaded code must first be analyzed by an interpreter before the desired act occurs, whereas object code, when executed, will perform the desired act straightaway (*see, e.g.*, definition of *interpreter*, at: http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?interpreter (last visited Aug. 30, 2004)).

Chow's generation and use of threaded code rather than object code is evidenced through such statements as "[t]he *threaded code* generated by the compiler is compact and efficient" (Chow, col. 8, lines 47-48), and "[t]he skeleton of the original input control statement then will be processed directly by the extended TCG (Threaded Code Generation) . . . to generate *threaded code* for this skeleton . . . " (Chow, col. 17, lines 39-43), among others. In so far as Chow generates interpreted, threaded code rather than object code in representing stored procedures, Chow does not meet the limitations of Applicant's claim 1. Therefore claim 1 and claims 2 through 14, which are dependent on claim 1, are patentable over Chow.

With respect to claim 15, as noted in the arguments above respecting claim 1, Chow does not teach generating "object code" corresponding to a stored procedure. As such, Chow does not meet the limitations of Applicant's claim 15. Therefore claim 15 and claims 16 through 22, which are dependent on claim 15, are patentable over Chow.

With respect to claim 23, Chow does not teach a database system comprising a plurality of "nodes" with an "evaluator module" in a first node and an "access module" in a second node. Rather, Chow teaches reducing communication overhead between a client and server through, among other things, executing compound statements "entirely" on the server side (*see* Chow, col. 3, lines 15-19). As interpreted by the Office, the client in Chow is a "first node" and the server is a "second node." However, since all evaluator and access functions in Chow are executed "entirely" on the server ("second node") side, Chow does not meet Applicant's limitation of having an evaluator module in a first node and an access module in a second node. Further, Chow does not suggest, let alone teach, the use of a controller in the first node adapted to execute a stored procedure "object code." Rather, as noted in the preceding arguments for claims 1 and 15, Chow teaches the use of threaded code rather than object code (*see* Chow, col. 8, lines 47-48). As such, Chow does not meet the limitations of Applicant's claim 23. Therefore, claim 23 and claims 24 through 26, which depend from claim 23, are patentable over Chow.

Finally, with respect to claim 27, Chow does not teach accessing "object code" in response to invocation of a stored procedure. Rather, as discussed above, Chow teaches the generation of threaded, as opposed to object, code to represent stored procedures, among other things. As such, Chow does not meet the limitations of Applicant's claim 27. Therefore, claim 27 and claims 28 and 29, which depend from claim 27, are patentable over Chow.

Conclusions

As listed and discussed above, Applicant has amended claims 14 and 17 to meet the requirements of 35 U.S.C. 112, second paragraph. Further, as shown above, Chow does not teach or suggest all of the limitations of Applicant's claims. As such, Applicant's claims are allowable over Chow.

In light of the foregoing amendments and arguments, Applicant asks the Office to reconsider this application and to allow all of the claims. Please apply any charges that might be due, excepting the issue fee but including fees for extensions of time, to deposit account 50-1673.

Respectfully,
John D. Cowart
John D. Cowart
Reg. No. 38,415

NCR Corporation
1700 South Patterson Blvd.
Dayton, Ohio 45479-0001

Tel. No. (858) 485-4903
Fax No. (858) 485-3255